PAIRWISE-TO-ALL: RELATIONAL EQUILIBRIUM FOR SIMULTANEOUS FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Relational Equilibrium State Estimation (RESE), a forecasting framework for multiple interacting systems that models how their relative relations evolve over time. RESE estimates these relations directly from observed trajectories, and the pairwise relations are weighted by reliability and reconciled into a consistent system-level equilibrium state. Individual forecasts are then recovered by combining this state with a separately estimated aggregate. Experiments on synthetic, exchange-rate, COVID-19, and multivariate datasets show that RESE is competitive on its own and can improve existing forecasting models. Standalone RESE achieves the lowest RMSE on the synthetic dataset, while most forecasting baselines improve when combined with RESE on exchange rates. RESE also remains stable under moderate noise, missing observations, and local shocks, and benefits from stronger cross-system dependence. Its computational cost remains low for small and moderate system sizes. Source and data are available at https://anonymous.4open.science/r/RESE-6024.

1 INTRODUCTION

Many forecasting problems involve several related systems observed over time, such as disease incidence across regions, demand across stores, traffic across sensors, or exchange rates across currencies. These systems do not evolve independently, and their past movements may contain useful information about one another. DeepAR shares information across related series through a global forecasting model Salinas et al. (2020). Crossformer models dependencies across variables Zhang & Yan (2023), while graph-based methods learn relations between individual series Cao et al. (2020); Shang et al. (2021). Equilibrium-based formulations provide another way to describe interacting components through a consistent system level state. In deep learning, deep equilibrium models represent network states as fixed points Bai et al. (2019). In forecasting, reconciliation methods enforce coherence across related series under aggregation constraints Hyndman et al. (2011), with extensions to probabilistic forecasts Rangapuram et al. (2021). Consistency has also been considered for learned cross-series structures Li et al. (2026b), while equilibrium-state forecasting has been used for simultaneous prediction across multiple systems Xu et al. (2026a).

Inspired by ESE, we introduce *Relational Equilibrium State Estimation* (RESE), which, unlike ESE, does not require attributes. RESE estimates equilibrium information from historical relations among the observed systems. For each pair of systems, RESE models its historical relation and forecasts the relation at the prediction horizon. Since these pairwise estimates are obtained separately, they are reconciled using reliability weights to form a common relative state for the whole system. This state is then combined with a separately forecast aggregate to recover the individual system forecasts. Relation estimation and aggregate forecasting are kept separate, allowing different forecasting models to be used within the same framework.

Our main contributions are:

- We introduce RESE to estimate the equilibrium state of multiple systems from their observed histories. It models time-varying cross-system relations directly, without requiring a predefined relation graph or aligned external attributes.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- RESE allows different system pairs to follow different relation dynamics and weights them by reliability. The pairwise relations are reconciled into a consistent equilibrium state, reducing the effect of local noise, missing data, and temporary shocks.
- RESE can be used alone or combined with different forecasting models, while remaining computationally efficient for small and moderate numbers of systems.

2 Relational Equilibrium in Multiple Systems

In a collection of peer systems, historical observations reveal persistent yet evolving relative relations among the systems, which contain information about their global equilibrium state. Because these pairwise relations are estimated separately, they may not be mutually consistent. RESE therefore reconciles them according to their reliability to obtain a globally consistent equilibrium state. We refer to this reliability-weighted equilibrium, inferred from the estimated relational structure, as the statistical equilibrium.

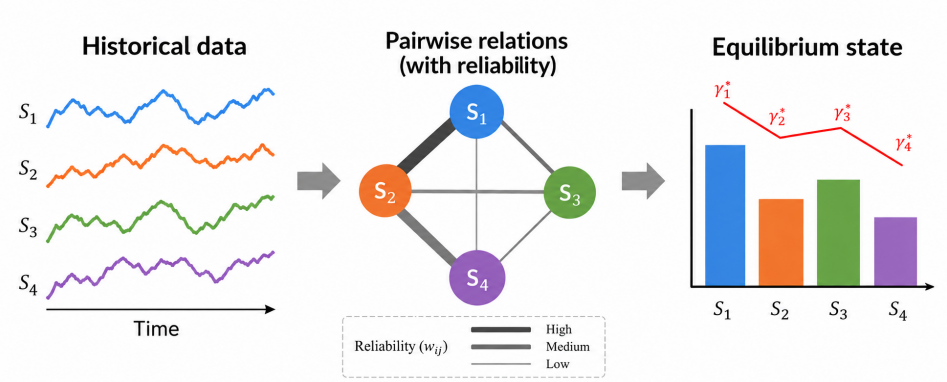

Figure 1: **Left:** Historical observations of four systems $S_1, \ldots, S_4$, where each trajectory represents the observed target $s_{i,t}$ of system S_i over time. **Middle:** The systems are represented as nodes in a pairwise relation graph. An edge between S_i and S_j represents the predicted relation $\widehat{r}_{ij,t}$, while its thickness indicates the corresponding reliability weight $w_{ij,t}$; thicker edges denote more reliable relation estimates. **Right:** The weighted pairwise relations are jointly reconciled through a latent relational representation and mapped to the equilibrium state $\Gamma_t^* = (\gamma_{1,t}^*, \ldots, \gamma_{N,t}^*)^\top$, where$\gamma_i^*$ which describes the relative configuration of the systems.

Multiple systems Let a multiple systems ensemble observed over time be denoted by

$$\mathcal{S} = \{S_1, \ldots, S_N\}, \qquad N \geq 2, \tag{1}$$

where each S_i represents a distinct system and $s_{i,t}$ denotes its observed target value at time t. We mainly consider peer systems, whose targets have the same or comparable meaning across systems.

System state The system state (ST) describes the observed relative allocation of the multiple system ensemble. At time t, let $s_{i,t}$ denote the observed value of system S_i, and define the aggregate as $M_t = \sum_{i=1}^{N} s_{i,t}$. The state of the system S_i is represented by its share of the aggregate,

$$\gamma_{i,t} = \frac{s_{i,t}}{M_t}, \qquad \sum_{i=1}^{N} \gamma_{i,t} = 1. \tag{2}$$

The system state is written as $\Gamma_t = (\gamma_{1,t}, \ldots, \gamma_{N,t})^\top$,which describes the relative configuration of the systems at time t.

Pairwise relations RESE extracts relational information from the observed trajectories themselves. For a pair of systems S_i and S_j, their relation at time t is represented as

$$r_{ij,t} = \mathcal{T}(s_{i,t}) - \mathcal{T}(s_{j,t}), \tag{3}$$

where the same transformation $\mathcal{T}(\cdot)$ is used across all systems. The collection of these relations forms the relation matrix $\mathbf{R}_t = [r_{ij,t}]_{N\times N}$. For each pair of systems, $r_{ij,t}$ forms a dynamic relational series describing how their relationship evolves over time. Based on these series, the time-varying cross-system structure can be characterized, from which the equilibrium state of multiple systems can be obtained.

Equilibrium state The global relational equilibrium describes the system-wide relative configuration implied jointly by the cross-system relations. We represent this equilibrium by the equilibrium state (ES), $\Gamma_t^* = (\gamma_{1,t}^*, \ldots, \gamma_{N,t}^*)^\top, \quad \sum_{i=1}^N \gamma_{i,t}^* = 1$, where $\gamma_{i,t}^*$ denotes the equilibrium share of system S_i. Unlike the system state (ST), which describes the observed relative allocation at time t, the equilibrium state represents the global relational configuration implied by the ensemble. While each pairwise relation describes the relative position of two systems, Γ_t^* characterizes their joint relational equilibrium.

3 Relational Equilibrium State Estimation

RESE treats the evolving cross-system relations as information about the relative state of the multiple systems ensemble. Each predicted pairwise relation provides information about the future relative positions of two systems, and the collection of these relations provides information about the state of the ensemble as a whole. Since the relations are estimated independently and with different levels of reliability, they may not imply exactly the same system state. RESE therefore reconciles them into a single equilibrium state, with more reliable relations receiving greater influence. This equilibrium is statistical and time-varying: it is inferred from the available historical relations rather than assumed to be a fixed balance. The resulting equilibrium state determines the relative allocation across systems and is combined with the aggregate forecast to obtain the final system-level forecasts.

3.1 Relation Representation

According to Equation 3, relations are defined through the transformation $\mathcal{T}$. The choice of $\mathcal{T}$ determines how differences between systems are represented.

For a given transformation, the observed system state can be represented in centered form as

$$u_{i,t} = \mathcal{T}(s_{i,t}) - \frac{1}{N}\sum_{k=1}^{N}\mathcal{T}(s_{k,t}), \tag{4}$$

Subtracting the cross-system mean centers the transformed values while preserving all pairwise differences. This removes the common level of the transformed system values and isolates their relative configuration of the systems and provides a common representation for estimating the equilibrium state from predicted relations. Specific representations are given in **Appendix A**.

3.2 Pairwise relation estimation and reliability

Once the relation representation is fixed, Equation 3 yields a historical relation series for each unordered pair (S_i, S_j). Each pairwise relation is then modeled separately using its historical series. Let $\mathcal{V}$ denote the historical time points used for validation. Each pairwise relation is fitted under several candidate specifications. For the k-th approach, the validation loss is

$$\mathcal{L}_{ij}^{(k)} = \frac{1}{|\mathcal{V}|}\sum_{t\in\mathcal{V}}\left(r_{ij,t} - \widehat{r}_{ij,t}^{(k)}\right)^2, \tag{5}$$

At each validation point, each pairwise relation is modelled using its preceding history.[1] For each pair, the approach with the lowest validation error is selected, and the corresponding error is denoted by $\mathcal{L}_{ij}^* = \min_k \mathcal{L}_{ij}^{(k)}$. This error is then used to determine the reliability weight:

$$w_{ij,t} = \exp\left[-\eta\frac{\sqrt{\mathcal{L}_{ij}^*}}{\mathrm{sd}(r_{ij})}\right], \qquad \eta \geq 0. \tag{6}$$

[1]The forecasting approaches used to model the historical relation dynamics are described in **Appendix B**.

where η controls the sensitivity of the weighting scheme to validation error. Lower-error relations receive larger weights in the equilibrium estimation.

3.3 Reliability-weighted equilibrium estimation

Since the pairwise relations are estimated separately, to obtain the equilibrium state, RESE jointly reconciles all estimated relations according to their reliability by solving

$$\widehat{\mathbf{u}}_t = \arg\min_{\mathbf{u}} \sum_{i<j} w_{ij,t} \left(u_{i,t} - u_{j,t} - \widehat{r}_{ij,t}\right)^2, \tag{7}$$

The pairwise relations determine only the relative positions of the systems, so the overall level of the coordinates is not identified. The common offset is therefore fixed by setting $\sum_{i=1}^{N} u_{i,t} = 0$. The weighted reconciliation then combines the pairwise relation estimates, where $w_{ij,t}$ controls the contribution of each relation and assigns greater influence to more reliable ones. [2]

3.4 Equilibrium State Estimation

The reliability-weighted reconciliation assigns one equilibrium coordinate to each system, giving $\widehat{\mathbf{u}}_t = (\widehat{u}_{1,t}, \ldots, \widehat{u}_{N,t})^\top$. Under the log-ratio representation, the difference between two coordinates determines their equilibrium share ratio, such that $\log(\gamma^*_{i,t}/\gamma^*_{j,t}) = \widehat{u}_{i,t} - \widehat{u}_{j,t}$. The equilibrium state is obtained by:

$$\gamma^*_{i,t} = \frac{\exp(\widehat{u}_{i,t})}{\sum_{k=1}^{N} \exp(\widehat{u}_{k,t})}, \tag{8}$$

3.5 Aggregate Forecasting and System Reconstruction

The equilibrium state determines the relative allocation across systems, while RESE predicts the overall change of the ensemble from the historical aggregate series. This change is represented by an aggregate trend factor θ_{t+h}, giving the RESE forecast

$$\widehat{\mathbf{s}}_{t+h} = \theta_{t+h} M_t \Gamma^*_{t+h}. \tag{9}$$

Here $\theta_{t+h} M_t$ gives the predicted aggregate, while Γ^*_{t+h} determines its allocation across systems. The estimation of θ_{t+h} is described in **Appendix D**. RESE can also be combined with an external forecasting model. In this case, the external model replaces the RESE aggregate predictor and provides $\widehat{M}_{t+h}$ directly. The system forecasts are then obtained as $\widehat{\mathbf{s}}_{t+h} = \widehat{M}_{t+h}\Gamma^*_{t+h}$, while the equilibrium state estimation remains unchanged.

4 Experiments

We evaluate RESE on the synthetic, exchange rate, and COVID-19 datasets, as these datasets contain multiple related systems. Then test RESE on standard multivariate forecasting benchmarks. Baseline results are taken from the cited studies. Further experiments examine RESE's components, sensitivity, and computational cost. [3].

4.1 Forecasting for Multiple Systems

We use synthetic data and two real-world datasets from Xu et al. (2026a), treating each currency or region as a system. The exchange rate data contain 16 currencies against the USD from November 11, 2019 to October 31, 2024. The COVID-19 data record daily new cases in Victoria, Australia,

[2] Cases in which unavailable pairwise relations lead to a disconnected relation graph are discussed in Appendix C.5.

[3] All RESE experiments are run using an AMD Ryzen 9 7950X CPU (16 cores, 4.50 GHz), 64 GB RAM, and an NVIDIA GeForce RTX 4090 GPU with 24 GB VRAM.

from January 1 to September 16, 2022, at three spatial scales: 20, 79, and 320 regions. We follow the experimental settings in that study and reuse its baseline and ESE results for comparison. [4]

Table 1: Results on the 10-system synthetic dataset with an input length of 20. Base denotes each forecasting baseline used alone; ESE and RESE use the corresponding baseline as their aggregate predictor. Standalone denotes each method using its built-in aggregate predictor. Time is measured in minutes. The best results are highlighted in bold, while the second-best results are underlined.

	Base			ESE			RESE (Ours)		
	RMSE	MAE	Time	RMSE	MAE	Time	RMSE	MAE	Time
Standalone	–	–	–	0.248	0.228	0.23	**0.239**	0.231	0.16
ARIMA	0.249	0.243	0.09	0.247	0.243	0.24	0.244	0.241	0.18
LSTM	0.263	0.216	2.47	0.263	<u>0.212</u>	0.48	0.255	0.227	0.39
DLinear	0.256	0.221	2.93	0.264	0.221	0.52	0.259	0.237	0.47
Informer	0.244	0.236	1.69	<u>0.241</u>	0.232	0.40	0.250	0.239	0.36
DeepAR	0.271	0.218	2.29	0.271	**0.210**	0.53	0.263	0.224	0.49
PatchTST	0.263	0.224	1.84	0.265	0.224	0.41	0.266	0.222	0.38

Synthetic Data We compare RESE and seven baselines: the attribute-based ESE; the statistical model ARIMA Hyndman & Athanasopoulos (2018); the recurrent models LSTM Hochreiter & Schmidhuber (1997) and DeepAR Salinas et al. (2020); the linear model DLinear Zeng et al. (2023); and the Transformer-based models Informer Zhou et al. (2021) and PatchTST Nie et al. (2023). Each baseline is evaluated on its own (Base) and as the aggregate predictor within ESE and RESE. The standalone row report the results of ESE and RESE without an external model. Table 1 shows that (1) standalone RESE achieves the lowest RMSE (0.239) among all configurations, with competitive MAE; (2) RESE maintains forecasting accuracy comparable to the other methods without requiring external system attributes as ESE; (3) RESE has lower reported runtimes than the neural baselines used alone and the corresponding ESE configurations. (4) ARIMA alone remains the fastest, owing to its low fitting cost in this small-scale setting.

Exchange rates The exchange rate experiments include twelve forecasting baselines, adding VAR Hyndman & Athanasopoulos (2018), NLinear Zeng et al. (2023), SCINet Liu et al. (2022a), TPGNN Liu et al. (2022b), and TimeLLM Jin et al. (2024) to the seven used on synthetic data. Table 2 presents one-step output with an input length of 100. As in ESE, we report both raw metrics, RMSE and MAE, and their normalized results, RMSE^* and MAE^*, to account for scale differences across currencies. Normalization details are given in **Appendix E**.

Table 2: Forecasting results on 16 exchange rates against the USD with an input length of 100. Base denotes each forecasting baseline used alone; ESE and RESE use the corresponding baseline as their aggregate predictor. Standalone denotes each method using its built-in aggregate predictor. RMSE^* and MAE^* are normalized metrics multiplied by 100. Time is measured in minutes. The best results are highlighted in bold, while the second-best are underlined.

	Base					ESE					RESE (Ours)				
	RMSE	MAE	RMSE^*	MAE^*	Time	RMSE	MAE	RMSE^*	MAE^*	Time	RMSE	MAE	RMSE^*	MAE^*	Time
Standalone	–	–	–	–	–	6.010	5.520	1.183	0.600	0.22	5.572	5.331	1.180	0.589	0.23
VAR	6.823	5.917	1.248	0.653	–	–	–	–	–	–	–	–	–	–	–
ARIMA	6.274	5.681	1.280	0.629	0.19	5.709	5.291	1.197	0.604	0.22	5.686	5.304	1.189	0.607	0.23
LSTM	6.168	5.668	1.218	0.604	5.71	5.621	5.257	1.190	0.601	0.58	5.607	5.238	1.179	0.597	0.59
DLinear	5.878	5.405	1.212	0.603	6.33	5.461	**5.102**	1.174	0.586	0.62	**5.435**	<u>5.117</u>	1.182	0.590	0.60
NLinear	5.963	5.459	1.214	0.608	6.71	5.518	5.171	1.179	0.593	0.64	5.499	5.139	1.181	0.597	0.65
Informer	6.012	5.585	1.219	0.608	3.58	5.563	5.204	1.182	0.595	0.45	5.547	5.119	1.194	0.587	0.43
SCINet	5.899	5.502	1.184	0.603	8.82	<u>5.481</u>	5.148	1.174	0.590	0.77	5.503	5.134	1.180	**0.583**	0.78
DeepAR	6.078	5.606	1.219	0.610	5.19	5.622	5.287	1.189	0.604	0.55	5.557	5.166	1.184	0.600	0.57
TPGNN	6.088	5.633	1.182	0.601	6.75	5.512	5.183	1.177	0.594	0.64	5.448	5.194	1.179	0.587	0.64
PatchTST	6.039	5.568	1.209	0.607	4.32	5.546	5.203	1.181	0.595	0.49	5.524	5.177	<u>1.172</u>	<u>0.583</u>	0.50
TimeLLM	5.913	5.524	1.182	0.603	4.53	5.487	5.186	**1.167**	0.599	0.54	5.492	5.181	1.178	0.584	0.56

Table 2 shows that (1) RESE alone gives competitive forecasts; (2) RESE improves the performance for most baselines; (3) accuracy is generally close to ESE despite RESE not using external system attributes; and (4) reported computational costs are close to those of ESE but remain much lower than those of most baselines used alone. VAR's computational cost is excluded because it requires manual parameter tuning.

COVID-19 Table 3 shows that (1) RESE gives accuracy close to ESE across all three spatial scales without using external system attributes; (2) all ten predictors used within RESE improve both

[4] Following Xu et al. (2026a), we use a 90:10 train-test split for models that require training.

RMSE and MAE over their standalone versions at 79 and 320 systems. Gains are less consistent at 20 systems, where PatchTST alone achieves the lowest errors; (3) reported computational cost remains lower than that of most baselines used alone at all levels of granularity; (4) RESE is faster than ESE at 20 systems but slower at 79 and 320 systems, as the number of pairwise relations grows. Table 3 reports the main results at the 79-city level. Additional results for 20 regions and 320 suburbs are provided in **Appendix F**.

Table 3: Forecasting results on the COVID-19 dataset with 79 cities and an input length of 100. Base denotes each forecasting baseline used alone; ESE and RESE use the corresponding baseline as their aggregate predictor. Time is measured in minutes.

	Base			ESE			RESE		
	RMSE	MAE	Time	RMSE	MAE	Time	RMSE	MAE	Time
Standalone	–	–	–	55.54	50.85	2.10	54.36	50.34	3.74
VAR	90.85	81.74	–	–	–	–	–	–	–
ARIMA	77.72	67.41	1.31	54.15	50.65	2.12	54.68	49.97	3.82
LSTM	61.42	57.26	39.96	55.42	49.93	2.59	54.79	50.22	4.22
DLinear	56.16	53.95	40.98	56.14	50.12	2.68	55.79	49.88	4.34
NLinear	55.62	52.23	45.67	55.14	51.33	2.70	54.86	51.42	4.47
Informer	62.01	60.57	26.97	55.48	49.23	2.40	54.67	49.45	4.03
SCINet	55.86	53.56	62.27	54.04	50.34	2.82	54.33	49.92	4.52
DeepAR	56.34	54.32	37.25	<u>50.99</u>	50.96	2.52	**50.36**	49.97	4.20
TPGNN	59.22	58.13	42.92	56.95	54.99	2.65	56.35	55.38	4.09
PatchTST	56.44	52.41	27.70	53.71	<u>47.25</u>	2.46	53.44	**46.84**	4.14
TimeLLM	59.48	52.64	24.15	51.66	47.84	2.50	51.84	48.35	4.62

These results suggest that RESE removes ESE's dependence on external attributes while retaining comparable forecasting accuracy and improving most baselines. RESE avoids attribute processing and can be less costly than ESE when the number of systems is small.

4.2 Multi-variate Benchmark

Although RESE was not developed specifically for multivariate time-series forecasting, it can be applied to this setting by treating each variable as a system, without requiring external attributes. We evaluate RESE on seven benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Electricity, Traffic, and Weather.[5] The input length is fixed at 96, with prediction horizons of 96, 192, 336, and 720. Performance is measured by MSE and MAE. [6]

Table 4 further evaluates RESE [7] on conventional multivariate time-series forecasting. Unlike ESE, which requires external attributes to estimate the equilibrium state, RESE removes this requirement and can therefore treat each variable in a multivariate time series as an individual system. Overall, RESE achieves competitive results across the seven benchmarks, although its advantage varies across datasets, horizons, and evaluation metrics. Specifically, (1) RESE performs well on ETTh1 and ETTm1 at shorter horizons, while remaining competitive on ETTh2 and ETTm2; (2) on Electricity, RESE achieves notably strong MAE results, although its MSE is less competitive; (3) on Traffic, the improvement is mainly observed in MSE at shorter horizons, whereas the MAE results are weaker; and (4) on Weather, RESE becomes more competitive at longer horizons and achieves the best MSE at $H = 720$. These results show that RESE can be extended to standard multivariate forecasting without changing its basic formulation. However, the gains are less consistent than in its original multi-system setting, suggesting that RESE is applicable to multivariate time series, but is not specifically optimised for this task.

[5]The baselines are GTR (Cao et al., 2026), TQNet (Lin et al., 2025), TimeXer (Wang et al., 2024b), CycleNet (Lin et al., 2024), SOFTS (Han et al., 2024), and PatchTST (Nie et al., 2023). The baseline results in Table 4 are taken from Cao et al. (2026).The benchmark datasets are available through `https://github.com/thuml/Time-Series-Library`.

[6]Full results with additional baselines are reported in **Appendix H**.

[7]To accommodate zero or negative observations in the multivariate time-series benchmarks, the signed-log transformation is adopted for the multivariate time-series benchmarks. Details are provided in **Appendix I.1**.

Table 4: Multivariate time-series forecasting results of RESE and six baseline methods across all prediction horizons. The look-back length T is fixed at 96, and the prediction horizons are 96, 192, 336, and 720. Performance is evaluated using MSE and MAE. The best results are highlighted in bold, while the second-best results are underlined.

Dataset	H	RESE (Ours)		GTR		TQNet		TimeXer		CycleNet		SOFTS		PatchTST	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.353**	**0.381**	0.367	0.391	0.371	0.393	0.382	0.403	0.375	0.395	0.381	0.399	0.414	0.419
	192	**0.392**	**0.384**	0.420	0.421	0.428	0.426	0.429	0.435	0.436	0.428	0.435	0.431	0.460	0.445
	336	**0.454**	0.449	0.476	0.447	0.476	**0.446**	0.468	0.448	0.496	0.455	0.480	0.452	0.501	0.466
	720	0.482	0.494	0.493	0.476	0.487	0.470	**0.469**	**0.461**	0.520	0.484	0.499	0.488	0.500	0.488
ETTh2	96	0.327	**0.334**	0.290	0.342	0.295	0.343	**0.286**	0.338	0.298	0.344	0.297	0.347	0.302	0.348
	192	**0.351**	**0.343**	0.362	0.389	0.367	0.393	0.363	0.389	0.372	0.396	0.373	0.394	0.388	0.400
	336	0.426	**0.412**	0.414	0.429	0.417	0.427	0.414	0.423	0.431	0.439	**0.410**	0.426	0.426	0.433
	720	0.437	0.433	0.423	0.442	0.433	0.446	**0.408**	**0.432**	0.450	0.458	0.411	0.433	0.431	0.446
ETTm1	96	**0.292**	**0.314**	0.305	0.349	0.311	0.353	0.318	0.356	0.319	0.360	0.325	0.361	0.329	0.367
	192	**0.335**	**0.370**	0.349	0.375	0.356	0.378	0.362	0.383	0.360	0.381	0.375	0.389	0.367	0.385
	336	0.394	0.427	**0.380**	**0.398**	0.390	0.401	0.395	0.407	0.389	0.403	0.405	0.412	0.399	0.410
	720	0.446	0.457	**0.435**	**0.436**	0.452	0.440	0.452	0.441	0.447	0.441	0.466	0.447	0.454	0.439
ETTm2	96	0.210	**0.218**	0.168	0.249	0.173	0.256	0.171	0.256	**0.163**	0.246	0.180	0.261	0.175	0.259
	192	0.241	**0.237**	0.232	0.292	0.238	0.298	0.237	0.299	**0.229**	0.290	0.246	0.306	0.241	0.302
	336	0.324	**0.327**	0.287	0.330	0.301	0.340	0.296	0.338	**0.284**	**0.327**	0.319	0.352	0.305	0.343
	720	0.394	0.399	**0.386**	**0.390**	0.397	0.396	0.392	0.394	0.389	0.391	0.405	0.401	0.402	0.400
Electricity	96	0.184	**0.176**	**0.134**	0.229	**0.134**	0.229	0.140	0.242	0.136	0.229	0.143	0.233	0.181	0.270
	192	0.193	**0.199**	**0.152**	0.245	0.154	0.247	0.157	0.256	**0.152**	0.244	0.158	0.248	0.188	0.274
	336	0.226	**0.217**	0.171	0.264	**0.169**	0.264	0.176	0.275	0.170	0.264	0.178	0.269	0.204	0.293
	720	0.230	0.298	0.208	0.300	**0.201**	**0.294**	0.211	0.306	0.212	0.299	0.218	0.305	0.246	0.324
Traffic	96	**0.333**	0.302	0.440	0.263	0.413	0.261	0.428	0.271	0.458	0.296	0.376	**0.251**	0.462	0.290
	192	**0.396**	0.374	0.454	0.274	0.432	0.271	0.448	0.282	0.457	0.294	0.398	**0.261**	0.466	0.290
	336	0.439	0.367	0.472	0.282	0.450	0.277	0.473	0.289	0.470	0.299	**0.415**	**0.269**	0.482	0.300
	720	0.475	0.393	0.514	0.301	0.486	0.295	0.516	0.307	0.502	0.314	**0.447**	**0.287**	0.514	0.320
Weather	96	0.236	0.217	**0.154**	**0.200**	0.157	**0.200**	0.157	0.205	0.158	0.203	0.166	0.208	0.177	0.210
	192	0.241	0.248	**0.202**	**0.244**	0.206	0.245	0.204	0.247	0.207	0.247	0.217	0.253	0.225	0.250
	336	0.294	**0.267**	**0.259**	0.287	0.262	0.287	0.261	0.290	0.262	0.289	0.282	0.300	0.278	0.290
	720	**0.312**	0.346	0.341	0.342	0.344	0.342	0.340	0.341	0.344	0.344	0.356	0.351	0.354	**0.340**

4.3 Analysis and Discussion

Table 5: Computational scaling of RESE with the number of systems. Relation estimation and global reconciliation are reported separately.

N	Pairs	Runtime (min)		Total
		Relation Est.	Reconciliation	
10	45	0.15	0.01	0.16
20	190	0.24	0.02	0.26
50	1225	2.22	0.02	2.24
100	4950	6.71	0.02	6.73
200	19900	18.59	0.03	18.62
300	44850	50.40	0.03	50.43

Computational Complexity For a complete relation graph with N systems, there are $\frac{N(N-1)}{2}$ unordered candidate pairs. After pairwise relation estimation, RESE performs a global reconciliation step using reliability weighting to estimate the equilibrium state. To distinguish these two sources of computation, Table 5 reports the runtime of pairwise relation estimation and global reconciliation separately as the number of systems increases. The main computational growth of the current complete-graph implementation comes from pairwise relation estimation, whose number of candidate pairs increases quadratically with N. The reconciliation step is performed after relation estimation and is reported separately to distinguish its cost from that of pairwise fitting. The current implementation of RESE is not linear in the number of systems. [8] Further implementation and complexity details are provided in **Appendix G.5**.

Robustness analysis We evaluate the robustness of RESE under three types of perturbation: (i) observation noise, (ii) missing observations, and (iii) heterogeneous local shocks. Noise and missing

[8] The scaling experiment uses the COVID-19 dataset. In this analysis, we focus only on computational runtime as the number of systems increases; forecasting accuracy is not considered.

data experiments are conducted on the 16 exchange rate series against the USD, while local shocks are evaluated on the 10-system synthetic dataset.

Table 6: Robustness analysis of RESE under observation noise and missing observationson the 16 exchange-rate series against the USD. Results are reported using RMSE^*.

(a) Observation noise

Noise	Affected systems		
	1	2	5
5%	1.18	1.18	1.17
10%	1.23	1.19	1.38
20%	1.28	1.34	1.53

(b) Missing observations

Missing	Affected systems		
	1	2	5
5%	1.17	1.19	1.20
10%	1.16	1.27	1.24
20%	1.22	1.29	1.49

Table 6 shows that RESE is relatively stable under moderate observation noise and missing data. When only one or two systems are affected, the changes in RMSE^* remain small across most settings. The effect becomes clearer when five systems are perturbed, especially at the 20% level. Missing observations show a similar trend, although the results are not strictly monotonic at lower levels. In general, RESE remains usable under moderate data degradation, but larger and more widespread perturbations lead to higher forecasting error.

We further test RESE under heterogeneous local shocks. The experiment is conducted on the 10-system synthetic dataset, where one system is temporarily perturbed to disrupt its historical relation with the others. We vary both the magnitude and duration of the shock and evaluate the resulting forecasting performance. The detailed perturbation setting is provided in Appendix J.1. Table 7 shows that RESE is relatively insensitive to mild local shocks. When $a = 0.1$, its error remains close to the unperturbed result across all shock durations. The effect becomes more visible for stronger shocks: with $a = 1.0$, the RMSE increases as the shock lasts longer, reaching 0.279 at $m = 100$. Despite this degradation, RESE remains competitive under stronger shocks. Overall, the results suggest that short or mild local deviations have limited impact on RESE, while stronger and more persistent shocks gradually weaken forecasting performance.

Table 7: RESE under heterogeneous local shocks on the 10-system synthetic dataset. a denotes the shock magnitude and m the shock duration; the first row is the unperturbed case.

m	a	RESE	DLinear	DLinear+RESE
–	–	0.239	0.256	0.259
10	0.1	0.241	0.253	0.257
10	1.0	0.252	0.254	0.265
50	0.1	0.247	0.259	0.251
50	1.0	0.264	0.278	0.269
100	0.1	0.258	0.257	0.255
100	1.0	0.279	0.282	0.274

Table 8: Forecasting results on the 10-system synthetic dataset under different levels of system dependence. ρ controls the strength of the common innovation, while $\hat{\rho}$ is the realized mean pairwise correlation between the generated target series.

ρ	$\hat{\rho}$	RESE	DLinear	DLinear+RESE
0.0	0.44	0.294	0.235	0.287
0.2	0.51	0.268	0.239	0.264
0.4	0.58	0.273	0.250	0.263
0.6	0.73	0.251	0.242	0.250
0.8	0.87	0.234	0.238	0.245

Effect of systems correlation RESE relies on information shared across systems, so its performance may depend on the strength of system dependence. To examine this effect, we conduct a controlled experiment on the 10-system synthetic dataset. The original data-generating process and system-specific parameters are kept fixed, while the amount of shared shock across systems is varied. Details of the construction are provided in **Appendix J.2**. We consider five levels of common shock strength, $\rho \in \{0, 0.2, 0.4, 0.6, 0.8\}$, and report the corresponding realized mean pairwise correlation $\hat{\rho}$. RESE, DLinear, and DLinear+RESE are evaluated under the same forecasting setting. Each setting is repeated over 10 random seeds, and the mean RMSE is reported.

Table 8 shows that RESE benefits from stronger system dependence. As the realized mean correlation increases from 0.44 to 0.87, the RMSE of RESE decreases from 0.294 to 0.234, while DLinear remains relatively stable. RESE is less competitive at lower correlation levels but becomes the

best-performing method at the highest level. This suggests that stronger cross-system dependence provides more useful relational information for RESE. Details are provided in **Appendix J.2**.

5 Related Work

Time-series forecasting Classical time-series forecasting is mainly based on statistical models that describe trend, seasonality, and temporal dependence explicitly Hyndman & Athanasopoulos (2018). Neural forecasting allows information to be shared across related series, as in DeepAR Salinas et al. (2020). Later work has paid particular attention to long-horizon forecasting. Informer was designed to handle long input sequences more efficiently Zhou et al. (2021). Autoformer introduces decomposition into the forecasting process Wu et al. (2021), while FEDformer models temporal patterns in the frequency domain Zhou et al. (2022). DLinear shows that simple linear models can still be competitive Zeng et al. (2023). PatchTST, in contrast, represents a time series using temporal patches Nie et al. (2023). Periodic and multiscale patterns are considered in TimesNet Wu et al. (2023) and TimeMixer Wang et al. (2024a). More recent studies have also explored pretrained and language-model-based forecasting, including Time-LLM Jin et al. (2024), AutoTimes Liu et al. (2024b), and related pretrained models Xiaoming et al. (2025).

Cross-series dependence For multivariate forecasting, performance also depends on how information is shared between variables. Crossformer models interactions across both time and variables Zhang & Yan (2023). iTransformer treats each variate history as a token and applies attention across variates Liu et al. (2024a). Another line of work represents cross-series dependence using graphs. StemGNN learns latent graph structure from the data Cao et al. (2020). Adaptive graph structures have also been used for multivariate forecasting Liu et al. (2022b), while discrete graph relations can be learned directly Shang et al. (2021). CrossGNN uses graph modeling to capture cross-scale interactions between variables Huang et al. (2023). SOFTS follows a different design, using a shared series core to collect information across variables and then redistribute it to individual series Han et al. (2024). These approaches use different structures, but all make cross-variable information part of the forecasting model.

Structured and time-varying relations Recent studies have looked more closely at how cross-variable relations are formed and how they change over time. TimeBridge models long-term cointegrated relations across variates Liu et al. (2025). VPNet limits cross-variate modeling to local interactions Liu et al. (2026). xCPD uses graph spectral decomposition to route channel–patch interactions dynamically Li et al. (2026a). Crisp builds sparse interaction structures according to spectral compatibility Wang et al. (2026). CoRA separates cross-channel correlations into time-varying and time-invariant parts Cheng et al. (2026). CPiRi learns content-dependent inter-channel relations while keeping the model invariant to channel ordering Xu et al. (2026b). These studies suggest that cross-series relations are often better viewed as selective and changing over time rather than as a single fixed structure.

6 Conclusion

We proposed RESE, a history-driven extension of ESE that estimates equilibrium information from relations among observed systems without requiring external attributes. Across synthetic, exchange-rate, COVID-19, and multivariate forecasting experiments, RESE shows competitive performance and can improve existing forecasting models. The additional analysis also shows that RESE is reasonably robust to data perturbations and benefits more when cross-system dependence is stronger.

Limitations RESE has two main limitations. (1) it relies on meaningful cross-system relations. When these relations are weak, unstable, or supported by limited history, the estimated equilibrium state may add little beyond individual series. (2) its cost grows with the number of systems because the number of pairwise relations increases quadratically.

Future work We will focus on reducing the cost of relation estimation. Parallel computation and neural implementations may allow multiple pairwise relations to be processed at the same time, making RESE more practical for larger systems.

AI USE STATEMENT

Generative AI tools were used only for grammar checking and language polishing. They were not used for dataset construction, methodology development, experimental design, code generation, result analysis, or scientific conclusions. All AI-assisted edits were reviewed and approved by the authors, who take full responsibility for the final manuscript.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used, including synthetic data, exchange-rate data, regional COVID-19 data, and standard multivariate time-series benchmarks, were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure that the results presented in this paper are reproducible. All code and datasets have been made publicly available in an anonymous repository to facilitate replication and verification. The experimental setup, including training steps, model configurations, and hardware details, is described in detail in the paper. We have also provided a full description of Relational Equilibrium State Estimation (RESE), including pairwise relation estimation, reliability weighting, equilibrium reconciliation, and forecast reconstruction, to assist others in reproducing our experiments.

REFERENCES

Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974. doi: 10.1109/TAC.1974.1100705.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 5 edition, 2015. ISBN 9781118675021.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17766–17778. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/cdf6581cb7aca4b7e19ef136c6e601a5-Paper.pdf.

Fanpu Cao, Lu Dai, Jindong Han, and Hui Xiong. Enhancing multivariate time series forecasting with global temporal retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QUJBPSfyui.

Hanyin Cheng, Xingjian Wu, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Cora: Boosting time series foundation models for multivariate forecasting through correlation-aware adapter. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ae3e173398d6e43fea63cbcc16fbaa98-Abstract-Conference.html.

Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997. ISBN 9780821803158.

Carl de Boor. *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Springer, 1978. doi: 10.1007/978-1-4612-6333-3.

James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, 2 edition, 2012. ISBN 9780199641178. doi: 10.1093/acprof:oso/9780199641178.001.0001.

Robert F. Engle and C. W. J. Granger. Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, 1987. doi: 10.2307/1913236.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2046.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21774–21797. PMLR, 2025. URL https://proceedings.mlr.press/v267/han25d.html.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. CrossGNN: Confronting noisy multivariate time series via cross interaction refinement. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2031.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Rob J. Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D. Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, 2008. ISBN 9783540719168. doi: 10.1007/978-3-540-71918-2.

Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational Statistics Data Analysis*, 55(9):2579–2589, 2011. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2011.03.006. URL https://www.sciencedirect.com/science/article/pii/S0167947311000971.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Søren Johansen. *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models*. Oxford University Press, 1995. ISBN 9780198774501. doi: 10.1093/0198774508.001.0001.

Dongyuan Li, Shun Zheng, Chang Xu, Jiang Bian, and Renhe Jiang. Routing channel-patch dependencies in time series forecasting with graph spectral decomposition. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/494ad6d9b36d9a1528dbf3baeb4e8a72-Abstract-Conference.html.

Zhengyu Li, Xiangfei Qiu, Yuhan Zhu, Xingjian Wu, Jilin Hu, Chenjuan Guo, and Bin Yang. Gcgnet: Graph-consistent generative network for time series forecasting with exogenous variables. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2f27964513a28d034530bfdd117ea31d-Abstract-Conference.html.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. CycleNet: Enhancing time series forecasting through modeling periodic patterns. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bfe7998398779dde03cad7a73b1f81b6-Abstract-Conference.html.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37797–37814. PMLR, 2025. URL https://proceedings.mlr.press/v267/lin25e.html.

Kun Liu, Renjun Jia, Ruifeng Yang, Xirui Zeng, Yuqi Liang, and Cen Chen. Are global dependencies necessary? scalable time series forecasting via local cross-variate modeling. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a832b65da26c4867a8581cb13d46ad6d-Abstract-Conference.html.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022a.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-Tao Xia. Timebridge: Non-stationarity matters for long-term time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39815–39840. PMLR, 2025. URL https://proceedings.mlr.press/v267/liu25cb.html.

Yijing Liu, Qinxian Liu, Jian-Wei Zhang, Haozhe Feng, Zhongwei Wang, Zihan Zhou, and Wei Chen. Multivariate time-series forecasting with temporal polynomial graph neural networks. *Advances in neural information processing systems*, 35:19414–19426, 2022b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024a.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *arXiv preprint arXiv:2402.02370*, 2024b.

Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 2005. ISBN 9783540401728. doi: 10.1007/978-3-540-27752-1.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Syama Sundar Rangapuram, Lucien D Werner, Konstantinos Benidis, Pedro Mercado, Jan Gasthaus, and Tim Januschowski. End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8832–8843. PMLR, 18–24 Jul 2021. URL http://proceedings.mlr.press/v139/rangapuram21a.html.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021.

Howell Tong. *Non-linear Time Series: A Dynamical System Approach*. Oxford University Press, 1990. ISBN 9780198522249. doi: 10.1093/oso/9780198522249.001.0001.

Binwu Wang, Gaoyun Lin, Jiaming Ma, Qihe Huang, Zhengyang Zhou, Xu Wang, Pengkun Wang, and Yang Wang. Crisp: A spectral-based interaction strategy for multivariate time series forecasting. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/63011.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/a7ac8a21e5a27e7ab31a5f42a0117bdb-Abstract-Conference.html.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. TimeXer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/0113ef4642264adc2e6924a3cbbdf532-Abstract-Conference.html.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is Mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025. doi: 10.1016/j.neucom.2024.129178.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Shi Xiaoming, Wang Shiyu, Nie Yuqi, Li Dianqi, Ye Zhou, Wen Qingsong, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *ICLR 2025: The Thirteenth International Conference on Learning Representations*. International Conference on Learning Representations, 2025.

Beinan Xu, Andy Song, Jiti Gao, and Feng Liu. Once-for-all: Scalable simultaneous forecasting via equilibrium state estimation. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2026a.

Jiyuan Xu, Wenyu Zhang, Xin Jing, Jiahao Nie, Shuai Chen, and Shuai Zhang. Cpiri: Channel permutation-invariant relational interaction for multivariate time series forecasting. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ebba182cb97864368fdb6ae00773a5e4-Abstract-Conference.html.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022.

A Relation Representation and Properties

Equation 3 defines the relation between systems S_i and S_j through a common transformation $\mathcal{T}$. Together with the centered coordinate in Equation 4, the pairwise relation satisfies

$$u_{i,t} - u_{j,t} = r_{ij,t}. \tag{10}$$

Thus centering removes only the common transformed level; it leaves all pairwise differences unchanged.

A.1 Log-ratio representation

For positive system values, the main RESE formulation uses $\mathcal{T}(s) = \log s$. Then

$$r_{ij,t} = \log \frac{s_{i,t}}{s_{j,t}} = \log \frac{\gamma_{i,t}}{\gamma_{j,t}}. \tag{11}$$

The second equality follows from $s_{i,t} = M_t \gamma_{i,t}$ and $s_{j,t} = M_t \gamma_{j,t}$. The common aggregate M_t cancels, so the relation contains information about relative allocation rather than the aggregate itself.

A common positive rescaling of all systems also leaves the log-ratio unchanged. For zero-valued non-negative observations, the implementation uses $\log(s+\epsilon)$ with a small $\epsilon > 0$. This is a numerical treatment of zeros rather than a change to the strictly positive log-ratio definition.

A.2 Recovery of equilibrium shares

Let $\widehat{\mathbf{u}}_t$ be the reconciled equilibrium coordinates from Equation 7. Under the log-ratio representation,

$$\widehat{u}_{i,t} - \widehat{u}_{j,t} = \log \frac{\gamma^*_{i,t}}{\gamma^*_{j,t}}.$$

Hence the equilibrium shares are proportional to $\exp(\widehat{u}_{i,t})$. Normalizing these terms to sum to one gives Equation 8. A common additive shift in all coordinates is removed by this normalization, so the recovered shares do not depend on the particular zero-sum representative used for $\widehat{\mathbf{u}}_t$.

A.3 Alternative relation coordinates

The graph reconciliation only requires pairwise differences under a common transformation. The reference implementation also contains the following coordinates:

Representation	$\mathcal{T}(s)$
log-ratio	$\log s$
asinh	$\operatorname{asinh}(s)$
signed-log	$\operatorname{sgn}(s)\log(1+\lvert s\rvert)$
additive	s

Only the log-ratio representation has the equilibrium-share identity in Equation 11. Under another transformation, $\mathbf{u}$ remains a reconciled relative coordinate, but its interpretation and reconstruction rule depend on the chosen transformation. The corresponding reconstruction is given in Appendix I.1.

B Pairwise Relation Models and Reliability

For each unordered pair (i, j), RESE constructs the historical relation series $\{r_{ij,t}$ and models it using observations available before the forecast origin. The same model library is available to every pair, but model selection is carried out separately. Different pairs may therefore use different relation models.

Every candidate ultimately returns the same quantity, $\widehat{r}_{ij,t+h}$. Model-specific coefficients are not inserted into the relation matrix; they are used only to produce this common relation forecast.

B.1 DEFAULT RELATION MODELS

The reference implementation uses six relation-model families:

Family	Relation dynamics represented
Stable	approximately constant relation
Local trend	gradually changing level and slope
Dynamic	autoregressive relation dynamics
Long-run equilibrium	joint bivariate dynamics with possible cointegration
Smooth nonlinear	smoothly varying nonlinear relation
Regime dependent	state-dependent relation dynamics

These families define the reference library rather than a restriction on the equilibrium solver. Another estimator can be used if it forecasts the same relation coordinate and is evaluated under the same validation protocol.

B.1.1 STABLE RELATION

The stable family represents a relation that fluctuates around a constant level,

$$r_{ij,t} = \mu_{ij} + e_{ij,t}, \tag{12}$$

where μ_{ij} is the relation level and $e_{ij,t}$ is a residual term. The reference implementation estimates the level by

$$\widehat{\mu}_{ij} = \operatorname{median}\left\{r_{ij,t}\right\}, \tag{13}$$

where L is the active lookback length. The resulting forecast is

$$\widehat{r}^{(\mathrm{S})}_{ij,t+h} = \widehat{\mu}_{ij}. \tag{14}$$

B.1.2 LOCAL-TREND RELATION

For relations whose level changes gradually, the reference implementation uses a local-linear-trend state-space model Durbin & Koopman (2012):

$$r_{ij,t} = \mu_{ij,t} + \varepsilon_{ij,t}, \tag{15}$$

$$\mu_{ij,t} = \mu_{ij,t-1} + \nu_{ij,t-1} + \eta_{ij,t}, \tag{16}$$

$$\nu_{ij,t} = \nu_{ij,t-1} + \zeta_{ij,t}. \tag{17}$$

Here $\mu_{ij,t}$ is the latent relation level, $\nu_{ij,t}$ is its local slope, and $\varepsilon_{ij,t}$, $\eta_{ij,t}$, and $\zeta_{ij,t}$ are the observation, level, and slope disturbances. The h-step relation forecast is the corresponding state-space forecast of the future level,

$$\widehat{r}^{(\mathrm{T})}_{ij,t+h} = \widehat{\mu}_{ij,t+h}. \tag{18}$$

B.1.3 DYNAMIC RELATION

The dynamic family models serial dependence in the relation using a low-order ARIMA model Box et al. (2015):

$$\Phi(B)(1-B)^d r_{ij,t} = \Theta(B) e_{ij,t}. \tag{19}$$

Here B is the backshift operator, d is the differencing order, and $\Phi(\cdot)$ and $\Theta(\cdot)$ are the autoregressive and moving-average polynomials. The reference search uses $p, q \leq 2$ and $d \leq 1$, where p and q are the orders of Φ and Θ. The order is selected by AIC Akaike (1974), and the fitted model returns $\widehat{r}^{(\mathrm{D})}_{ij,t+h}$.

B.1.4 LONG-RUN EQUILIBRIUM RELATION

This family models the two transformed system trajectories jointly. The term "equilibrium" in this subsection refers to a possible long-run relation within one pair; it is distinct from the system-wide equilibrium state estimated by RESE.

Define the bivariate transformed series

$$\mathbf{x}_{ij,t} = \begin{bmatrix} \mathcal{T}(s_{i,t}) \\ \mathcal{T}(s_{j,t}) \end{bmatrix}. \tag{20}$$

The implementation first tests the two transformed trajectories for cointegration using an Engle–Granger test Engle & Granger (1987). If cointegration is supported, a rank-one VECM is fitted Johansen (1995); Lütkepohl (2005):

$$\Delta \mathbf{x}_{ij,t} = \mathbf{\Pi}_{ij} \mathbf{x}_{ij,t-1} + \sum_{\ell=1}^{p-1} \mathbf{\Gamma}_{ij,\ell} \Delta \mathbf{x}_{ij,t-\ell} + \mathbf{c}_{ij} + \boldsymbol{\varepsilon}_{ij,t}. \tag{21}$$

Here $\mathbf{\Pi}_{ij}$ is the long-run coefficient matrix, $\mathbf{\Gamma}_{ij,\ell}$ are short-run coefficient matrices, $\mathbf{c}_{ij}$ is a deterministic term, and $\boldsymbol{\varepsilon}_{ij,t}$ is the innovation vector.

If cointegration is not supported, the same family uses a bivariate VAR Lütkepohl (2005):

$$\mathbf{x}_{ij,t} = \mathbf{c}_{ij} + \sum_{\ell=1}^{p} \mathbf{A}_{ij,\ell} \mathbf{x}_{ij,t-\ell} + \boldsymbol{\varepsilon}_{ij,t}, \tag{22}$$

where $\mathbf{A}_{ij,\ell}$ are the VAR coefficient matrices. The lag order is selected using AIC. Both VECM and VAR forecast the two transformed system coordinates, after which the common relation forecast is recovered as

$$\widehat{r}^{(\mathrm{E})}_{ij,t+h} = \widehat{x}_{i,t+h} - \widehat{x}_{j,t+h}. \tag{23}$$

B.1.5 Smooth nonlinear relation

The smooth nonlinear family models the relation trajectory with a B-spline basis de Boor (1978):

$$r_{ij,t} = \beta_{ij,0} + \sum_{m=1}^{K_B} \beta_{ij,m} B_m(t) + e_{ij,t}, \tag{24}$$

where $B_m(t)$ is the m-th spline basis function, $\beta_{ij,m}$ are its coefficients, and K_B is the number of basis terms. The coefficients are estimated with ridge regularization Hoerl & Kennard (1970):

$$\widehat{\boldsymbol{\beta}}_{ij} = \arg\min_{\boldsymbol{\beta}} \left[\sum_{t} \left(r_{ij,t} - \beta_0 - \sum_{m=1}^{K_B} \beta_m B_m(t) \right)^2 + \lambda_{\mathrm{N}} \|\boldsymbol{\beta}\|_2^2 \right], \tag{25}$$

where $\lambda_{\mathrm{N}} \geq 0$ is the ridge penalty. The fitted smooth function is extrapolated to obtain the future relation. Outside the observed range, the reference implementation uses linear extrapolation of the spline basis.

B.1.6 Regime-dependent relation

The regime-dependent family uses a two-regime SETAR(1) model Tong (1990):

$$r_{ij,t} = \begin{cases} a_{ij,1} + b_{ij,1} r_{ij,t-1} + e_{ij,t}, & r_{ij,t-1} \leq c_{ij}, \\ a_{ij,2} + b_{ij,2} r_{ij,t-1} + e_{ij,t}, & r_{ij,t-1} > c_{ij}. \end{cases} \tag{26}$$

Here c_{ij} is the regime threshold and $(a_{ij,r}, b_{ij,r})$ are the intercept and autoregressive coefficient in regime $r \in \{1, 2\}$. Candidate thresholds are taken from empirical quantiles of the lagged relation. The threshold with the smallest in-sample squared residual error is retained, and multi-step forecasts are generated recursively.

B.2 Historical validation and model selection

The main text gives the validation loss in Equation 5. In the horizon-matched implementation, let o denote a validation origin and let $\mathcal{O}_h$ be the set of origins that have an observed target h steps ahead. The loss for candidate k is

$$\mathcal{L}^{(k)}_{ij,h} = \frac{1}{|\mathcal{O}_h|} \sum_{o \in \mathcal{O}_h} \left(r_{ij,o+h} - \widehat{r}^{(k)}_{ij,o+h|o} \right)^2. \tag{27}$$

Thus candidate selection is based on historical forecasts made at the same horizon as the final forecast. When a fixed lookback L is used, the same lookback is applied at every validation origin.

The selected family and its validation error are

$$\widehat{k}_{ij,h} = \arg\min_{k} \mathcal{L}^{(k)}_{ij,h}, \qquad \mathcal{L}^{*}_{ij,h} = \mathcal{L}^{(\widehat{k}_{ij,h})}_{ij,h}. \tag{28}$$

Here $\widehat{k}_{ij,h}$ denotes the selected candidate for pair (i, j) at horizon h. The selected model is then refitted on the relation history available at the forecast origin.

B.3 Reliability weight

The validation error is converted to a reliability weight as described in Equation 6. For numerical stability, the benchmark implementation uses the scale

$$s_{ij} = \max\left\{\mathrm{sd}(r_{ij}), 1.4826\ \mathrm{MAD}(r_{ij}), 10^{-4}, \epsilon\right\}, \tag{29}$$

where $\mathrm{MAD}(r_{ij})$ is the median absolute deviation of the relation series and $\epsilon > 0$ is the numerical tolerance. The corresponding benchmark weight is

$$w_{ij,t} = \exp\left(-\eta \frac{\sqrt{\mathcal{L}^{*}_{ij,h}}}{s_{ij}}\right), \qquad \eta \geq 0. \tag{30}$$

The compact expression in the main text uses $\mathrm{sd}(r_{ij})$ directly; Equation 29 records the numerical safeguards used in the benchmark implementation.

Each retained pair therefore enters the equilibrium step as the pair $(\widehat{r}_{ij,t}, w_{ij,t})$. Since $r_{ji,t} = -r_{ij,t}$ and $r_{ii,t} = 0$, only one triangular half of the relation matrix contains independent information.

C Equilibrium Reconciliation and Identification

If all estimated pairwise relations were exactly compatible, there would be a single coordinate vector whose differences reproduced every retained relation. Because the pairwise models are fitted independently, exact compatibility need not hold. The weighted reconciliation in the main text finds one common coordinate vector while giving more influence to relations with larger reliability weights.

C.1 Graph form of the reconciliation

Let $\mathcal{E}$ be the set of retained unordered pairs. Choose one orientation for each edge in $\mathcal{E}$, and let $\mathbf{B} \in \mathbb{R}^{|\mathcal{E}| \times N}$ be the corresponding incidence matrix. Let $\mathbf{r}$ collect the estimated relations in that orientation, and let $\mathbf{W} = \mathrm{diag}(w_e)$ be the diagonal matrix of edge weights. Equation 7 is then equivalent to

$$\widehat{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \mathbf{W}^{1/2}(\mathbf{B}\mathbf{u} - \mathbf{r}) \right\|_2^2. \tag{31}$$

The associated weighted graph Laplacian and right-hand side are

$$\mathbf{L} = \mathbf{B}^{\top}\mathbf{W}\mathbf{B}, \qquad \mathbf{b} = \mathbf{B}^{\top}\mathbf{W}\mathbf{r}. \tag{32}$$

Here $\mathbf{L}$ is used only as a compact representation of the linear system implied by the weighted least-squares problem.

C.2 Connected relation graphs

For a connected weighted graph with positive edge weights, $\mathbf{L}$ has a one-dimensional nullspace spanned by the all-ones vector Chung (1997). The relation coordinates are therefore identified up to one common additive constant. Imposing $\mathbf{1}^{\top}\mathbf{u} = 0$ gives a unique representative.

The constrained normal equations can be written as

$$\begin{bmatrix} \mathbf{L} & \mathbf{1} \\ \mathbf{1}^{\top} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}, \tag{33}$$

where λ is the Lagrange multiplier associated with the zero-sum constraint. This matrix equation is a computational form of the constrained weighted least-squares solution, not an additional equilibrium definition.

C.3 Cycle consistency

If exact node coordinates exist, the implied pairwise differences sum to zero around every closed cycle:

$$r_{i_1 i_2} + r_{i_2 i_3} + \cdots + r_{i_K i_1} = 0. \tag{34}$$

This follows directly by substituting $r_{ij} = u_i - u_j$, because the node coordinates cancel around the cycle.

For a graph with E retained edges, N nodes, and C connected components, the number of independent cycles is $E - N + C$ Chung (1997). When this number is positive, cycles provide redundant constraints and the least-squares residual measures disagreement among them. In a forest there is no such redundancy, so a zero residual can occur by construction.

C.4 Complete unweighted graph

If all unordered pairs are present with equal weights, the zero-sum solution has the closed form

$$\widehat{u}_i = \frac{1}{N} \sum_{j=1}^{N} r_{ij}. \tag{35}$$

Equation 35 follows directly from the normal equations of the complete unweighted version of Equation 31. Under the log-ratio representation, Equation 8 then converts these coordinates into equilibrium shares.

C.5 Disconnected relation graphs

If some relations are unavailable or screened out, the retained graph may contain $C > 1$ connected components, $\mathcal{C}_1, \ldots, \mathcal{C}_C$. Relations identify positions within each component but not the relative offsets between components.

Let $\mathbf{u}_t^{\mathrm{ST}}$ be the centered coordinate of the observed system state from Equation 4. Define $\mathbf{A} \in \mathbb{R}^{C \times N}$ by

$$A_{c,i} = \begin{cases} 1, & i \in \mathcal{C}_c, \\ 0, & \text{otherwise.} \end{cases} \tag{36}$$

The component offsets are anchored by

$$\mathbf{A}\mathbf{u} = \mathbf{A}\mathbf{u}_t^{\mathrm{ST}}. \tag{37}$$

The c-th row of this constraint preserves the centered-coordinate sum of component $\mathcal{C}_c$. It fixes only the offsets that are not identified by the retained relations and does not change any within-component edge difference.

D Aggregate Forecasting and System Reconstruction

The equilibrium state determines relative allocation across systems, while the aggregate series is forecast separately. Let $\widehat{M}_{t+h}$ be the aggregate forecast based on observations available through time t. In the notation of the main text,

$$\widehat{M}_{t+h} = \theta_{t+h} M_t. \tag{38}$$

When $M_t > 0$, this gives $\theta_{t+h} = \widehat{M}_{t+h}/M_t$.

D.1 Aggregate predictor library

The reference implementation compares four predictors of the same aggregate series.

Persistence. The persistence forecast is

$$\widehat{M}^{(\mathrm{P})}_{t+h} = M_t. \tag{39}$$

Random walk with drift. Let t_0 be the first time point in the active fit window. The average drift over that window is

$$\widehat{d}_t = \frac{M_t - M_{t_0}}{t - t_0}, \tag{40}$$

and the corresponding forecast is

$$\widehat{M}^{(\mathrm{D})}_{t+h} = M_t + h\widehat{d}_t. \tag{41}$$

Damped Holt trend. The damped-trend exponential-smoothing model follows the standard Holt formulation Hyndman et al. (2008). Let ℓ_t be the estimated level, b_t the estimated trend, and $\phi_M \in (0, 1]$ the damping parameter. Its forecast is

$$\widehat{M}^{(\mathrm{H})}_{t+h} = \ell_t + \left(\sum_{q=1}^{h} \phi_M^q\right) b_t. \tag{42}$$

The level, trend, smoothing coefficients, and damping parameter are estimated from the active aggregate history.

Low-order ARIMA. The ARIMA aggregate predictor uses the same standard ARIMA form as Equation 19, now applied to M_t Box et al. (2015). The reference search uses $p, q \le 2$ and $d \le 1$, with the within-family order selected by AIC Akaike (1974).

D.2 Aggregate model selection

Let $\mathcal{O}_h$ again denote the historical validation origins with an observed aggregate h steps ahead. For aggregate predictor k,

$$\mathcal{L}^{(k)}_{M,h} = \frac{1}{|\mathcal{O}_h|} \sum_{o \in \mathcal{O}_h} \left(M_{o+h} - \widehat{M}^{(k)}_{o+h|o}\right)^2. \tag{43}$$

The selected aggregate family is

$$\widehat{k}_{M,h} = \arg\min_k \mathcal{L}^{(k)}_{M,h}. \tag{44}$$

The selected model is refitted on the aggregate history available at time t. If that refit fails, the next validation-ranked model that can be fitted is used. In the non-negative log-ratio setting, the reference implementation clips the final aggregate forecast at zero if necessary.

D.3 System reconstruction

For standalone RESE, the final system forecast follows the aggregate–allocation form inherited from ESE Xu et al. (2026a):

$$\widehat{s}_{i,t+h} = \widehat{M}_{t+h}\gamma^*_{i,t+h} = \theta_{t+h} M_t \gamma^*_{i,t+h}. \tag{45}$$

Here $\gamma^*_{i,t+h}$ is the equilibrium share estimated by RESE.

When RESE is combined with an external forecasting model, only the aggregate forecasting step is replaced:

$$\widehat{s}_{i,t+h} = \widehat{M}^{\mathrm{ext}}_{t+h}\gamma^*_{i,t+h}. \tag{46}$$

Pairwise relation estimation, reliability weighting, and equilibrium-state reconciliation remain unchanged.

E NORMALIZED RMSE AND MAE FOR EXCHANGE RATES

As in ESE (Xu et al., 2026a), we normalize each currency's forecasting errors by its mean exchange rate and then average across currencies:

$$\mathrm{RMSE}^* = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{RMSE}_i}{\bar{s}_i}, \qquad \mathrm{MAE}^* = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathrm{MAE}_i}{\bar{s}_i}. \tag{47}$$

Here, N is the number of currencies, RMSE_i and MAE_i are the forecasting errors for currency i, and $\bar{s}_i$ is its mean observed exchange rate. This normalization reduces the influence of scale differences across currencies. Lower values indicate better forecasting accuracy.

F ADDITIONAL COVID-19 RESULTS

We report the additional COVID-19 forecasting results at the 20-region and 320-suburb levels in Tables 9 and 10, respectively. The 79-city results are reported in the main paper. Base denotes each forecasting model used alone, while ESE and RESE use the corresponding model as the aggregate predictor. Time is measured in minutes. For RMSE and MAE, the best results are shown in bold and the second-best results are underlined.

Table 9: Forecasting results on the COVID-19 dataset with 20 regions and an input length of 100.

	Base			ESE			RESE		
	RMSE	MAE	Time	RMSE	MAE	Time	RMSE	MAE	Time
Standalone	–	–	–	84.54	83.91	1.31	86.37	84.45	0.28
VAR	95.79	90.86	–	–	–	–	–	–	–
ARIMA	79.70	79.59	0.33	75.31	73.64	1.33	76.06	73.97	0.29
LSTM	78.45	74.96	9.23	72.79	69.47	1.80	73.15	68.94	0.76
DLinear	79.47	72.41	10.40	73.61	<u>66.94</u>	1.86	72.99	67.46	0.81
NLinear	74.96	70.77	11.56	71.64	69.98	1.87	72.20	68.76	0.79
Informer	76.84	71.79	5.95	73.43	71.48	1.66	74.03	71.35	0.64
SCINet	79.75	69.94	14.18	71.84	69.71	2.04	71.73	70.02	1.01
DeepAR	83.41	74.65	9.22	74.82	72.91	1.77	75.22	72.67	0.72
TPGNN	78.37	69.86	10.58	72.24	70.64	1.81	72.16	69.82	0.79
PatchTST	**70.01**	**63.52**	6.92	<u>70.74</u>	66.97	1.79	71.17	67.13	0.85
TimeLLM	73.47	70.02	6.20	71.11	67.01	1.39	72.03	67.42	0.73

As shown in Table 9, RESE improves most standalone predictors at the 20-region level, although the gains are less consistent and standalone PatchTST gives the lowest RMSE and MAE. At the 320-suburb level, Table 10 shows that all ten predictors combined with RESE improve both RMSE and MAE over their standalone versions. The larger system size also leads to a substantial increase in relation-estimation time because many more pairwise relations must be estimated.

G BENCHMARK PROTOCOL AND COMPUTATIONAL DETAILS

This section records the protocol used in the benchmark experiments. These choices concern historical windows, preprocessing, relation screening, sequence outputs, and computation. They do not change the definition of the equilibrium state.

Table 10: Forecasting results on the COVID-19 dataset with 320 suburbs and an input length of 100.

	Base			ESE			RESE		
	RMSE	MAE	Time	RMSE	MAE	Time	RMSE	MAE	Time
Standalone	–	–	–	4.83	4.58	2.23	4.89	4.61	56.84
VAR	8.07	7.68	–	–	–	–	–	–	–
ARIMA	7.44	6.24	5.07	4.74	4.41	2.30	4.69	4.37	56.89
LSTM	6.23	5.47	149.26	4.68	4.37	2.72	4.74	4.36	58.74
DLinear	5.93	5.10	163.57	4.51	4.34	2.87	4.63	4.47	59.22
NLinear	5.47	4.99	160.09	4.47	**4.18**	2.86	<u>4.44</u>	4.22	58.31
Informer	6.05	5.80	100.08	4.60	4.37	2.62	4.56	4.41	58.73
SCINet	5.94	5.59	206.06	4.46	4.24	2.94	**4.33**	4.31	58.57
DeepAR	6.29	5.04	130.67	4.79	4.46	2.73	4.88	4.50	58.74
TPGNN	6.06	5.41	158.84	4.68	4.24	2.86	4.67	<u>4.19</u>	59.13
PatchTST	5.11	4.43	109.89	4.51	4.36	2.66	4.46	4.29	57.99
TimeLLM	5.12	4.82	107.73	4.62	4.31	2.83	4.59	4.28	59.46

G.1 Forecast windows and leakage control

Let o denote a validation or test origin. The implementation supports an expanding history or a sliding history of fixed length L:

$$\mathcal{H}_o = \begin{cases} \{1, \dots, o\}, & \text{expanding history,} \\ \{o - L + 1, \dots, o\}, & \text{sliding history.} \end{cases} \tag{48}$$

Here $\mathcal{H}_o$ denotes the observations available for fitting at origin o. When a fixed L is used, the same lookback is applied at every legal origin and in the final refit.

The implementation also allows a validation stride, a limit on the number of recent validation origins, and a minimum successful-fit rate. Model-family selection is completed using pre-test data. During test evaluation, the selected relation and aggregate families remain fixed; only their parameters are refitted from the history available at the current test origin.

G.2 Preprocessing

If preprocessing is used, its parameters are estimated from the training portion and then held fixed. For example, train-only standardization for system i is

$$\widetilde{s}_{i,t} = \frac{s_{i,t} - \mu_i^{\text{train}}}{\sigma_i^{\text{train}}}, \tag{49}$$

where μ_i^{train} and σ_i^{train} are the training mean and standard deviation. Relation estimation is performed in the working coordinate $\widetilde{s}$. After system reconstruction, the inverse preprocessing map is applied before evaluation. The identity map corresponds to forecasting in the original data space.

G.3 Persistence screening used in the benchmark

The reference implementation can compare a candidate relation model with a persistence forecast evaluated on the same origins and horizon:

$$\widehat{r}^{(\mathrm{P})}_{ij,o+h|o} = r_{ij,o}. \tag{50}$$

Let $\mathcal{L}^{(\mathrm{P})}_{ij,h}$ be the persistence validation loss. The relative improvement of candidate k is

$$\Delta^{(k)}_{ij,h} = \frac{\mathcal{L}^{(\mathrm{P})}_{ij,h} - \mathcal{L}^{(k)}_{ij,h}}{\mathcal{L}^{(\mathrm{P})}_{ij,h} + \epsilon}. \tag{51}$$

With the default zero margin, a candidate must have strictly smaller validation loss than persistence. If a positive margin δ is specified, the candidate must satisfy $\Delta^{(k)}_{ij,h} \geq \delta$.

If the highest-ranked candidate fails during final refitting, a lower-ranked candidate is used only if it independently passes the same persistence screen. A rejected pair is omitted from the retained graph. A relation forecast equal to zero remains a valid relation and is not treated as missing.

G.4 Sequence-output forecasting

For a forecast block ending at horizon H, the reference implementation selects the relation model at the endpoint horizon H. After refitting, the selected model returns the path

$$(\widehat{r}_{ij,t+1}, \ldots, \widehat{r}_{ij,t+H}) . \tag{52}$$

The aggregate predictor returns an analogous aggregate path. Equilibrium reconciliation and system reconstruction are then applied at each step $h = 1, \ldots, H$. The selection step is not repeated separately for every intermediate horizon.

G.5 Computational cost

A complete relation graph contains

$$E_{\max} = \frac{N(N-1)}{2} \tag{53}$$

unordered pairs. If K relation models are evaluated at R validation origins, the pairwise selection stage requires

$$O(N^2 K R) \tag{54}$$

candidate fits, apart from model-specific fitting costs. Fits for different pairs and validation origins are independent and can be parallelized. The current reference implementation executes these statistical fits serially.

If K_M aggregate predictors are compared over R_M aggregate validation origins, aggregate selection adds $O(K_M R_M)$ fits and does not contain the quadratic pairwise factor.

After relation estimation, let E denote the number of retained edges. The equilibrium solve uses an $E \times N$ incidence matrix. The reference implementation uses dense graph arrays and a constrained dense solve. A sparse incidence matrix and sparse Laplacian representation require $O(E + N)$ structural storage and can be used when the retained graph is large.

Persistence screening reduces the number of edges passed to the equilibrium solver, but it occurs after candidate validation. It therefore does not reduce the initial cost of evaluating the complete pairwise candidate graph.

H Full Multivariate Forecasting Results

Table 11 reports the complete multivariate forecasting results, including the five additional baselines omitted from the main table for readability: S-Mamba (Wang et al., 2025), RAFT (Han et al., 2025), TimeMixer (Wang et al., 2024a), iTransformer (Liu et al., 2024a), and DLinear (Zeng et al., 2023). All experimental settings are identical to those reported in the main paper.

I Optional Equilibrium Extensions

The main formulation uses the reliability-weighted equilibrium directly. The reference code also contains optional extensions. They are kept separate from the core estimator.

I.1 Reconstruction under alternative relation coordinates

The main-text reconstruction uses equilibrium shares and is therefore specific to the log-ratio representation. Under another monotone, invertible transformation, let q_i denote the reconciled coordinate of system i and let $\widehat{M}$ denote the forecast aggregate. Reconstruction takes the form

$$\widehat{s}_i = \mathcal{T}^{-1}(q_i + \kappa), \tag{55}$$

Table 11: Full multivariate time-series forecasting results of RESE and eleven baseline methods across all prediction horizons. The look-back length T is fixed at 96, and the prediction horizons are 96, 192, 336, and 720. Performance is evaluated using MSE and MAE. The best results are highlighted in bold, while the second-best results are underlined.

Dataset	H	RESE (Ours)		GTR		RAFT		S-Mamba		TQNet		TimeXer		CycleNet		SOFTS		TimeMixer		iTransformer		PatchTST		DLinear	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.353**	**0.381**	0.367	0.391	0.376	0.401	0.386	0.405	0.371	0.393	0.382	0.403	0.375	0.395	0.381	0.399	0.375	0.400	0.386	0.405	0.414	0.419	0.386	0.400
	192	**0.392**	**0.384**	0.420	0.421	0.417	0.424	0.443	0.437	0.428	0.426	0.429	0.435	0.436	0.428	0.435	0.431	0.429	0.421	0.441	0.436	0.460	0.445	0.437	0.432
	336	**0.454**	0.449	0.476	0.447	**0.454**	**0.441**	0.489	0.468	0.476	0.446	0.468	0.448	0.496	0.455	0.480	0.452	0.484	0.458	0.487	0.458	0.501	0.466	0.481	0.459
	720	0.482	0.494	0.493	0.476	**0.465**	0.465	0.502	0.489	0.487	0.470	0.469	**0.461**	0.520	0.484	0.499	0.488	0.498	0.482	0.503	0.491	0.500	0.488	0.519	0.516
ETTh2	96	0.327	**0.334**	0.290	0.342	0.289	0.342	0.296	0.348	0.295	0.343	**0.286**	0.338	0.298	0.344	0.297	0.347	0.289	0.341	0.297	0.349	0.302	0.348	0.333	0.387
	192	**0.351**	**0.343**	0.362	0.389	0.382	0.400	0.376	0.396	0.367	0.393	0.363	0.389	0.372	0.396	0.373	0.394	0.372	0.392	0.380	0.400	0.388	0.400	0.477	0.476
	336	0.426	**0.412**	0.414	0.429	0.424	0.439	0.424	0.431	0.417	0.427	0.414	0.423	0.431	0.439	0.410	0.426	**0.386**	0.414	0.428	0.432	0.426	0.433	0.594	0.541
	720	0.437	0.433	0.423	0.442	0.433	0.457	0.426	0.444	0.433	0.446	**0.408**	**0.432**	0.450	0.458	0.411	0.433	0.412	0.434	0.427	0.445	0.431	0.446	0.831	0.657
ETTm1	96	**0.292**	**0.314**	0.305	0.349	0.325	0.369	0.333	0.368	0.311	0.353	0.318	0.356	0.319	0.360	0.325	0.361	0.320	0.357	0.334	0.368	0.329	0.367	0.345	0.372
	192	**0.335**	**0.370**	0.349	0.375	0.364	0.388	0.376	0.390	0.356	0.378	0.362	0.383	0.360	0.381	0.375	0.389	0.361	0.381	0.377	0.391	0.367	0.385	0.380	0.389
	336	0.394	0.427	**0.380**	**0.398**	0.391	0.407	0.408	0.413	0.390	0.401	0.395	0.407	0.389	0.403	0.405	0.412	0.390	0.404	0.426	0.420	0.399	0.410	0.413	0.413
	720	0.446	0.457	**0.435**	**0.436**	0.444	0.438	0.475	0.448	0.452	0.440	0.452	0.441	0.447	0.441	0.466	0.447	0.454	0.441	0.491	0.459	0.454	0.439	0.474	0.453
ETTm2	96	0.210	**0.218**	0.168	0.249	0.176	0.264	0.179	0.263	0.173	0.256	0.171	0.256	**0.163**	0.246	0.180	0.261	0.175	0.258	0.180	0.264	0.175	0.259	0.193	0.292
	192	0.241	**0.237**	0.232	0.292	0.241	0.306	0.250	0.309	0.238	0.298	0.237	0.299	**0.229**	0.290	0.246	0.306	0.237	0.299	0.250	0.309	0.241	0.302	0.284	0.362
	336	0.324	**0.327**	0.287	0.330	0.302	0.345	0.312	0.349	0.301	0.340	0.296	0.338	**0.284**	**0.327**	0.319	0.352	0.298	0.340	0.311	0.348	0.305	0.343	0.369	0.427
	720	0.394	0.399	**0.386**	**0.390**	0.404	0.405	0.411	0.406	0.397	0.396	0.392	0.394	0.389	0.391	0.405	0.401	0.391	0.396	0.412	0.407	0.402	0.400	0.554	0.522
Electricity	96	0.184	**0.176**	**0.134**	0.229	0.152	0.257	0.139	0.235	**0.134**	0.229	0.140	0.242	0.136	0.229	0.143	0.233	0.153	0.247	0.148	0.240	0.181	0.270	0.197	0.282
	192	0.193	**0.199**	**0.152**	0.245	0.156	0.257	0.159	0.255	0.154	0.247	0.157	0.256	**0.152**	0.244	0.158	0.248	0.166	0.256	0.162	0.253	0.188	0.274	0.196	0.285
	336	0.226	**0.217**	0.171	0.264	0.171	0.272	0.176	0.272	**0.169**	0.264	0.176	0.275	0.170	0.264	0.178	0.269	0.185	0.277	0.178	0.269	0.204	0.293	0.209	0.301
	720	0.230	0.298	0.208	0.300	0.221	0.300	0.204	0.298	**0.201**	**0.294**	0.211	0.306	0.212	0.299	0.218	0.305	0.225	0.310	0.225	0.317	0.246	0.324	0.245	0.333
Traffic	96	**0.333**	0.302	0.440	0.263	0.388	0.265	0.382	0.261	0.413	0.261	0.428	0.271	0.458	0.296	0.376	**0.251**	0.462	0.285	0.395	0.268	0.462	0.290	0.650	0.396
	192	**0.396**	0.374	0.454	0.274	0.400	0.278	**0.396**	0.267	0.432	0.271	0.448	0.282	0.457	0.294	0.398	**0.261**	0.473	0.296	0.417	0.276	0.466	0.290	0.598	0.370
	336	0.439	0.367	0.472	0.282	**0.411**	0.287	0.417	0.276	0.450	0.277	0.473	0.289	0.470	0.299	0.415	**0.269**	0.498	0.296	0.433	0.283	0.482	0.300	0.605	0.373
	720	0.475	0.393	0.514	0.301	0.456	0.305	0.460	0.300	0.486	0.295	0.516	0.307	0.502	0.314	**0.447**	**0.287**	0.506	0.313	0.467	0.302	0.514	0.320	0.645	0.394
Weather	96	0.236	0.217	**0.154**	**0.200**	0.185	0.241	0.165	0.210	0.157	**0.200**	0.157	0.205	0.158	0.203	0.166	0.208	0.163	0.209	0.174	0.214	0.177	0.210	0.196	0.255
	192	0.241	0.248	**0.202**	**0.244**	0.237	0.287	0.214	0.252	0.206	0.245	0.204	0.247	0.207	0.247	0.217	0.253	0.208	0.250	0.221	0.254	0.225	0.250	0.237	0.296
	336	0.294	**0.267**	0.259	0.287	0.290	0.327	0.274	0.297	0.262	0.287	0.261	0.290	0.262	0.289	0.282	0.300	**0.251**	0.287	0.278	0.296	0.278	0.290	0.283	0.335
	720	**0.312**	0.346	0.341	0.342	0.367	0.380	0.350	0.345	0.344	0.342	0.340	0.341	0.344	0.344	0.356	0.351	0.339	0.341	0.358	0.349	0.354	**0.340**	0.345	0.381

where the common offset κ is chosen to satisfy

$$\sum_{i=1}^{N} \mathcal{T}^{-1}(q_i + \kappa) = \widehat{M}. \tag{56}$$

If $\mathcal{T}^{-1}$ is continuous and strictly increasing, the left-hand side is strictly increasing in κ. The solution is therefore unique whenever the requested aggregate lies in the attainable range.

For the additive representation, the solution is explicit:

$$\kappa = \frac{\widehat{M} - \sum_i q_i}{N}, \qquad \widehat{s}_i = q_i + \kappa. \tag{57}$$

For asinh and signed-log coordinates, the reference implementation solves the one-dimensional aggregate constraint numerically.

I.2 Finite-horizon adjustment toward equilibrium

The estimated equilibrium can be treated as a horizon-specific target rather than requiring the forecast state to reach it immediately. Let $\mathbf{u}_t^{\mathrm{ST}}$ be the current centered coordinate and $\widehat{\mathbf{u}}_{t+h}^{*}$ the equilibrium coordinate for horizon h. The adjusted state is

$$\widehat{\mathbf{u}}_{t+h}^{(\alpha)} = (1 - \alpha_h)\mathbf{u}_t^{\mathrm{ST}} + \alpha_h \widehat{\mathbf{u}}_{t+h}^{*}, \qquad 0 \le \alpha_h \le 1. \tag{58}$$

Here α_h is the fraction of the current-to-equilibrium gap removed by horizon h. The special cases $\alpha_h = 0$ and $\alpha_h = 1$ give current-state persistence and the full-equilibrium RESE forecast, respectively. A per-step persistence parameter $\rho \in [0, 1]$ can alternatively be used through $\alpha_h = 1 - \rho^h$.

I.3 Robust reweighting of inconsistent relations

For fixed relation forecasts and fixed weights, the weighted least-squares solution is already determined. The optional robust refinement instead keeps the relation forecasts fixed and updates only their working weights according to their global residuals.

For retained edge e, let B_e denote the corresponding row of the incidence matrix, let r_e be its relation forecast, and let w_e^{pred} be its original validation weight. At iteration q,

$$\mathbf{u}^{(q)} = \arg\min_{\mathbf{u}} \sum_e w_e^{(q)} (B_e \mathbf{u} - r_e)^2, \tag{59}$$

$$e_e^{(q)} = r_e - B_e \mathbf{u}^{(q)}, \tag{60}$$

$$w_e^{(q+1)} = w_e^{\text{pred}}\, g(e_e^{(q)}). \tag{61}$$

Here $e_e^{(q)}$ is the residual for edge e, and $g(\cdot)$ is a robust downweighting function. The default implementation uses the Huber rule Huber (1964),

$$g(e) = \begin{cases} 1, & |e| \le c, \\ c/|e|, & |e| > c, \end{cases} \tag{62}$$

where $c > 0$ is the residual cutoff. The relation forecasts themselves are not moved toward the reconciled state.

Because this refinement changes the weights after historical validation, results obtained with it should be reported separately from the core RESE estimator.

J Additional Analysis

J.1 Heterogeneous Local Shocks

To examine the sensitivity of RESE to temporary system-specific disturbances, we perturb one system in the 10-system synthetic dataset over a local time interval. The perturbed observations are defined as

$$y'_t = y_t \left[1 + a \sin\left(\frac{\pi}{2}(t - t_0) + \frac{\pi}{4}\right)\right], \qquad t_0 \le t < t_0 + m, \tag{63}$$

where $a \in \{0.1, 1.0\}$ controls the shock magnitude and $m \in \{10, 50, 100\}$ controls the shock duration. All other systems remain unchanged. This setting temporarily disturbs the historical relation between the perturbed system and the remaining systems, allowing us to evaluate the stability of RESE when local relational structure is disrupted.

J.2 Controlled system Correlation

To examine the effect of system dependence, we retain the original synthetic data-generating process and modify only the dependence among its innovation terms. For system i, the original target process is

$$y_{i,t} = \log\left(C_i + \beta y_{i,t-1} + \epsilon_{i,t}\right), \tag{64}$$

where $\beta = 1.2$ and C_i is sampled from $[50, 100]$, following the original synthetic-data setting. For the correlation experiment, the same C_i and β are retained across all settings.

We introduce a common and a system-specific component into the innovation:

$$\epsilon_{i,t}^{(\rho)} = \sqrt{\rho}\, z_t + \sqrt{1 - \rho}\, z_{i,t}, \qquad \rho \in [0, 1], \tag{65}$$

where z_t is shared across all systems and $z_{i,t}$ is independently generated for each system. Both components follow the same innovation setting as the original synthetic data. The resulting process is

$$y_{i,t}^{(\rho)} = \log\left(C_i + \beta y_{i,t-1}^{(\rho)} + \epsilon_{i,t}^{(\rho)}\right). \tag{66}$$

The parameter ρ controls the amount of common disturbance across systems. At $\rho = 0$, the systems are driven by independent innovations; as ρ increases, a larger fraction of the innovation is shared.

Innovation variance. Let the common and system-specific innovation components have zero mean and the same variance σ^2. Since they are independent,

$$\begin{aligned}\operatorname{Var}\left(\epsilon_{i,t}^{(\rho)}\right) &= \rho \operatorname{Var}(z_t) + (1-\rho)\operatorname{Var}(z_{i,t}) \\ &= \rho\sigma^2 + (1-\rho)\sigma^2 \\ &= \sigma^2. \end{aligned} \tag{67}$$

Thus, varying ρ changes the dependence across systems without changing the marginal innovation variance.

Cross-system dependence. For two different systems $i \neq j$,

$$\begin{aligned}\operatorname{Cov}\left(\epsilon_{i,t}^{(\rho)}, \epsilon_{j,t}^{(\rho)}\right) &= \rho \operatorname{Var}(z_t) \\ &= \rho\sigma^2, \end{aligned} \tag{68}$$

and therefore, before the nonlinear target transformation,

$$\operatorname{Corr}\left(\epsilon_{i,t}^{(\rho)}, \epsilon_{j,t}^{(\rho)}\right) = \rho. \tag{69}$$

Because the target series are generated recursively through the nonlinear mapping in Eq. 66, the resulting correlation between the target series does not necessarily equal ρ. We therefore use ρ as the dependence-control parameter and report the realized mean pairwise target correlation,

$$\hat{\rho} = \frac{2}{N(N-1)} \sum_{i<j} \widehat{\operatorname{Corr}}(y_i, y_j), \tag{70}$$

computed from the training portion of each generated dataset.This construction keeps the original synthetic forecasting mechanism and system-specific parameters fixed while varying the amount of shared innovation across systems. It therefore provides a controlled setting for examining how RESE responds to different levels of system dependence.